

Description

TECHNICAL FIELD

[0001] The present disclosure generally relates to the field of point cloud data sources. More particularly, it deals with point cloud compression (PCC). Thus, the disclosure concerns a method for encoding a point cloud and a corresponding encoder. It also concerns a method for reconstructing a point cloud and a corresponding decoder. It further concerns computer programs implementing the encoding and reconstructing methods of the invention.

BACKGROUND

[0002] The approaches described in this section could be pursued, but are not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated herein, the approaches described in this section are not prior art to the claims in this application and are not admitted to be prior art by inclusion in this section.

[0003] A point cloud consists of a set of points. Each point is defined by its spatial location (x, y, z), i.e. geometry information in the (X,Y,Z) coordinate system, or any other coordinate system, such as the polar system, and different attributes, which typically include the color information in (R, G, B) or (Y, U, V) or any other color coordinate system.

[0004] Point cloud data sources are found in many applications. Important applications relying on huge point cloud data sources can be found in geographic information systems, robotics, medical tomography and scientific visualization.

[0005] Beyond these applications that are more industrial and scientifically oriented, the rise in popularity of inexpensive 3D scanners based on time of flight or other depth sensing technologies, 3D capturing on mobile devices and the rise of cloud based 3D printing are creating a huge demand for large scale interoperable compressed 3D point cloud storage and transmission data formats in the consumer market.

[0006] Scanned 3D point clouds often have thousands of points and occupy large amounts of storage space. Additionally, they can be generated at a high rate when captured live from 3D scanners, increasing the storage requirements even further. Therefore, point cloud compression is critical for efficient networked distribution and storage.

[0007] Different compression methods of point clouds have been proposed in the prior art.

[0008] The first compression method is based on the octree based point-cloud representation. It is described, for instance, in J. Peng and C.-C. Jay Kuo, "Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition," ACM Trans. Graph, vol. 21, no. 2, pp. 609-616, July 2005.

[0009] An octree is a tree data structure where every branch node represents a certain cube or cuboid bounding volume in space. Starting at the root, every branch has up to eight children, one for each sub-octant of the node's bounding box.

[0010] An example is shown in Figure 1 and the left half of Figure 2.

[0011] For compression, a single bit is used to mark whether every child of a branch node is empty and then this branch node configuration can be efficiently represented in a single byte, assuming some consistent ordering of the eight octants. By traversing the tree in breadth-first order and outputting every child node configuration byte encountered, the point distribution in space can be efficiently encoded. Upon reading the encoded byte stream, the number of bits set in the first byte tells the decoder the number of consecutive bytes that are direct children. The bit positions specify the voxel/child in the octree they occupy. The right half side of Figure 2 illustrates this byte stream representation for the octree in the left half.

[0012] Due to its hierarchical nature, the octree representation is very efficient in exploiting the sparsity of point clouds. Thus, octree decomposition based strategy is very efficient for compressing sparse point clouds. However, octree-based representation is inefficient for representing and compressing dense point clouds.

[0013] The second compression method is based on the segmentation based point cloud representation. It is described, for instance, in T. Ochotta and D. Saupe, "Compression of point-based 3d models by shape-adaptive wavelet coding of multi-heightfields," in Proc. Eurographics Symp. on Point-Based Graphics, 2004, and in J. Digne, R. Chaine, S. Valette, et al, "Self-similarity for accurate compression of point sampled surfaces," Computer Graphics Forum, vol. 33, p. 155-164, 2014.

[0014] The segmentation based point cloud representation comprises three steps: plane-like decomposition, plane projection and coding. In the first step, the point cloud data is segmented into plane-like patches. Then the resultant patches are projected onto one or several planes. In the last step, the projected image(s) are compressed. Efficient image/video compression techniques can be used for compressing the projected map(s).

[0015] Plane-like segmentation based PCC has been proved to be very efficient at coding dense point clouds which represent approximate piecewise linear surfaces. However, there are several disadvantages of segmentation-based PCC. Indeed, this method requires resampling onto a regular grid, which introduces approximation errors. Also, it is necessary to generate many patches for complex shapes which implies a high computation complexity for complicated

texture.

[0016] There are works trying to combine the advantages of the above two representations of point clouds such as J. K. N. Blodow, R. Rusu, S. Gedikli and E. S. M Beetz, "Real-time compression of point cloud streams," in Robotics and Automation (ICRA), 2012 IEEE International Conference on , pp.778,785, 14-18 May, 2012.

[0017] These works first decompose the input point clouds in the manner of octree decomposition. After each cuboid division operation, it is checked whether the points falling into the same newly generated leaf cuboid can be approximated by a plane. If the answer is yes, the division of the corresponding cuboid is stopped and the related points are encoded by projecting on the corresponding plane. Then, the positions of the plane-like patches can be efficiently represented and compressed by the octree method. However, both the scale and position of the resultant plane-like patches are constrained by the octree decomposition step. The piece-wise linear characteristics of the point cloud cannot be fully exploited.

SUMMARY

[0018] The present disclosure proposes a solution for improving the situation.

[0019] Accordingly, the present invention provides a method for encoding a point cloud comprising:

- segmenting the point cloud into first plane-like patches;
- projecting the first plane-like patches to projected planes using inverse projection parameters;
- encoding the projected planes;
- determining sparse points in the point cloud; and
- encoding the sparse points using an octree-decomposition.

[0020] Thus, the encoding method of the present disclosure combines the segmentation and octree compression techniques. By first segmenting the point cloud, its piece-wise linearity can be fully exploited using the plane-like patches. At the same time, the encoder also takes advantage from the high efficiency of the octree based point cloud compression when compressing a sparsely distributed point cloud. This is achieved through the use of the octree compression only for the encoding of the sparse points still not encoded after the segmentation.

[0021] Advantageously, the sparse points include points not belonging to any plane-like patch.

[0022] These points not belonging to any plane-like patch are called high-curvature points.

[0023] According to an embodiment, the encoding method includes:

- reconstructing second plane-like patches from the encoded projected planes;
- comparing the second plane-like patches and the first plane-like patches in order to obtain, for each point in the first plane-like patches, a reconstruction error; and
- comparing, for each point in the first plane-like patches, the reconstruction error to a threshold.

[0024] Advantageously, the sparse points include residual points for which the reconstruction error is larger than the threshold.

[0025] By including the residual points in the sparse points which are encoded using the octree decomposition technique, the encoding is improved.

[0026] Advantageously, the inverse projection parameters include positions and/or orientations of the plane-like patches.

[0027] According to an embodiment, the encoding method includes encoding the inverse projection parameters.

[0028] Advantageously, the encoding method includes decoding the encoded inverse projection parameters.

[0029] Preferably, projecting the first plane-like patches to projected planes uses the decoded inverse projection parameters.

[0030] The present invention also provides an encoder for encoding a point cloud, comprising:

- a segmentation module configured to segment the point cloud into first plane-like patches;
- a projection module configured to project the first plane-like patches to projected planes using inverse projection parameters;
- a first coding module configured to encode the projected planes;
- a computation module configured to determine sparse points in the point cloud; and
- a second coding module configured to encode the sparse points using an octree-decomposition.

[0031] These encoder modules are advantageously implemented by one or more processors within the encoder.

[0032] Advantageously, the first coding module is configured to encode and decode the inverse projection parameters.

[0033] The present disclosure also provides a method for reconstructing a point cloud, comprising:

- receiving an encoded bitstream including encoded projected planes and encoded sparse points;
- decoding the received encoded bitstream;
- reconstructing plane-like patches of the point cloud.

[0034] Advantageously, the encoded bitstream includes encoded inverse projection parameters and the reconstructing method includes decoding the encoded inverse projection parameters.

[0035] The present disclosure also provides a decoder for reconstructing a point cloud, including:

- a receiver configured to receive an encoded bitstream including encoded projected planes and encoded sparse points;
- a decoding module configured to decode the received encoded bitstream; and
- a reconstruction module configured to reconstruct plane-like patches of the point cloud.

[0036] These decoder modules are advantageously implemented by one or more processors within the decoder.

[0037] The methods according to the disclosure may be implemented in software on a programmable apparatus. They may be implemented solely in hardware or in software, or in a combination thereof.

[0038] Since the present invention can be implemented in software, the present invention can be embodied as computer readable code for provision to a programmable apparatus on any suitable carrier medium. A carrier medium may comprise a storage medium such as a floppy disk, a CD-ROM, a hard disk drive, a magnetic tape device or a solid state memory device and the like.

[0039] The disclosure thus provides a computer-readable program comprising computer-executable instructions to enable a computer to perform the encoding method of the invention.

[0040] The diagram of figure 3 illustrates an example of the general algorithm for such computer program.

[0041] The disclosure also provides a computer-readable program comprising computer-executable instructions to enable a computer to perform the reconstructing method of the invention.

[0042] The diagram of figure 5 illustrates an example of the general algorithm for such computer program.

BRIEF DESCRIPTION OF THE DRAWINGS

[0043] The present invention is illustrated by way of examples, and not by way of limitation, in the figures of the accompanying drawings, in which like reference numerals refer to similar elements and in which:

- Figure 1, already described, is a schematic view illustrating an octree-based point cloud representation according to the prior art;
- Figure 2 already described, is a schematic view illustrating an overview of an octree data structure;
- Figure 3 is a flowchart showing the steps of encoding a point cloud, according to an embodiment of the present disclosure;
- Figure 4 shows an example of projection of the plane-like patches, according to an embodiment of the present disclosure;
- Figure 5 is a flowchart showing the steps of reconstructing a point cloud, according to an embodiment of the present disclosure;
- Figure 6 is a schematic view illustrating an encoder, according to an embodiment of the invention; and
- Figure 7 is a schematic view illustrating a decoder, according to an embodiment of the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0044] The method of encoding a point cloud according to an embodiment of the present invention is illustrated in the flowchart of Figure 3.

[0045] The 3D point cloud is first segmented at step 2 into patches which can be approximated by planes, called as plane-like patches.

[0046] According to an embodiment, the segmentation uses the partitioning described in T. Ochotta and D. Saupe, "Compression of point-based 3d models by shape-adaptive wavelet coding of multi-heightfields" in Proc. Eurographics Symp. on Point-Based Graphics, 2004.

[0047] Another example is as follows.

For ($i = 0; i < \text{LOOP}_{\text{NUM}}; i++$)

{

Randomly select N points of the point cloud;

Build N plane-like patches, each of which only includes one of the selected N points;

For each patch, add the neighboring points of the points of the patch which don't belong to any plane-like patch, if the patch is still close enough to a plane after the new point is added;

Try to combine patches by checking for each patch, whether or not, after combining the patch with each of its neighboring patches, the new patch is still close enough to a plane. If yes, combine the two patches;

Calculate the reconstruction error between the reconstructed point cloud and the original point cloud;

};

where $\text{LOOP}_{\text{NUM}} = \alpha_{\text{Loop}} * \#Pt, N = \alpha_N * \#Pt$, where $\#Pt$ is the number of the points in the point cloud, α_{Loop} and α_N are predefined thresholds. For instance, it is possible to choose $\alpha_{\text{Loop}}=0.1$ and $\alpha_N=0.2$.

[0048] Then, the segmentation results, i.e. the plane-like patches with the minimum reconstruction error are selected.

[0049] An error metric can be used to judge whether a subset of points of the point cloud can be approximated by a plane. The points not belonging to any plane-like patches are called high-curvature points.

[0050] Then, at step 4, inverse projection parameters of the plane-like patches are determined. These parameters include the positions and/or orientations of the patches.

[0051] At step 6, the inverse projection parameters are encoded.

[0052] Advantageously, as the patch positions are usually sparsely distributed in the 3D space, the octree decomposition based compression method can efficiently compress the patch positions. The other parameters can be compressed by quantization and entropy coding which are known compression techniques in the art.

[0053] According to an embodiment, the patch orientations are encoded using the method described in US 9,214,042.

[0054] The resulting encoded inverse projection parameters are included in the encoded bitstream transmitted to the decoder.

[0055] At step 8, the encoded inverse projection parameters are decoded.

[0056] At step 10, the first plane-like patches are projected to projected planes using the decoded inverse projection parameters. The projection can be achieved by aligning the object coordinate system of each plane-like patch with the world coordinate system.

[0057] Figure 4 shows a 2D example in order to explain an embodiment for implementing the projection.

[0058] According to this embodiment, the centroid of each patch is translated to the origin of the XYZ coordinate system, and then the orientation axes of each patch are rotated to be aligned with the world coordinate system. The object coordinate system of each plane-like patch can be determined using principle component analysis (PCA). Then, each transformed patch is projected to XOY plane.

[0059] Then the projected planes are encoded at step 12 using any suitable patch coding method, such as the method described in T. Ochotta and D. Saupe, "Compression of point-based 3d models by shape-adaptive wavelet coding of multi-heightfields" in Proc. Eurographics Symp. on Point-Based Graphics, 2004.

[0060] The resulting data of the encoded projected planes are included in the encoded bitstream transmitted to the

decoder.

[0061] Then, the encoded projected planes are decoded at step 14.

[0062] The decoded projected planes are used together with the decoded inverse projection parameters, at step 16, to reconstruct plane-like patches. Using the decoded patch orientations, the plane-like patches are first rotated to make the X, Y and Z axes paralleled with the three orientation axes. Then, the rotated patches are translated to make the origin (0, 0, 0) overlapped with the decoded patch position.

[0063] Then, at step 18, the original plane-like patches, obtained at step 2, are compared with the reconstructed plane-like patches, obtained at step 16. A reconstruction error of each point belonging to the original plane-like patches is calculated.

[0064] The points with reconstruction errors larger than a threshold ERR preferably corresponding to a user required reconstruction error are considered as residual points.

[0065] At step 20, the high-curvature points and the residual points, forming the set of sparse points, are encoded using an octree decomposition based compression method such as the method described in J. Peng and C.-C. Jay Kuo, "Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition," ACM Trans. Graph, vol. 21, no. 2, pp. 609-616, July 2005.

[0066] The data resulting from the octree coding in step 20 are included in the encoded bitstream transmitted to the decoder.

[0067] According to a preferred embodiment, the encoded bitstream includes:

- A header, containing the number of the plane-like patches, lengths of each compressed projected plane, length of the compressed positions of the plane-like patches, depth of the octree used to compress the plane-like patch positions, the bounding box of all plane-like patch positions, the length of the compressed orientations of plane-like patches, the quantization parameter of the orientations of the plane-like patches, the length of the compressed sparse points, the depth of the octree used to compress the sparse points and the bounding box of all the sparse points,
- The compressed projected plane-like patches one by one, sorted by the decoding order of the plane-like patch positions,
- The compressed plane-like patch positions,
- The compressed plane-like patch orientations one by one, sorted by the decoding order of the plane-like patch positions,
- The compressed sparse points.

[0068] Figure 5 shows the steps of reconstructing a point cloud, i.e. the decoding steps implemented by the decoder after receiving the encoded bitstream including the encoded inverse projection parameters, the encoded projected planes and the encoded sparse points.

[0069] At step 22, the projected plane-like patches are decoded.

[0070] Then, at step 24, the inverse projection parameters are decoded.

[0071] According to a preferred embodiment, the plane-like patch positions are reconstructed at step 26 using the length of the compressed positions of the plane-like patches, the depth of the octree used to compress the plane-like patch positions and the bounding box of all plane-like patch positions. Then, the plane-like patch orientations are reconstructed using the length of the compressed orientations of plane-like patches and the quantization parameter of the orientations of the plane-like patches.

[0072] The sparse points, comprising the high-curvature points and the residual points, are decoded at step 28 using the length of the compressed sparse points, the depth of the octree used to compress the sparse points and the bounding box of the sparse points.

[0073] The decoding steps 22 and 24 are exchangeable and they may also be performed simultaneously with the step 28.

[0074] The resulting decoded patches and decoded sparse points form the decoded point cloud.

[0075] Figure 6 is a block diagram of an exemplary embodiment of an encoder 30 implementing the encoding method of the present disclosure.

[0076] Advantageously, the encoder 30 includes one or more processors and a memory 32.

[0077] The encoder 30 includes:

- a segmentation module 34 configured to segment the point cloud into first plane-like patches;
- a projection module 36 configured to project the first plane-like patches to projected planes using inverse projection parameters;
- a first coding module 38 configured to encode the projected planes;
- a computation module 39 configured to determine sparse points in the point cloud; and
- a second coding module 40 configured to encode the sparse points using an octree-decomposition.

[0078] According to a preferred embodiment, the first coding module is configured to encode and decode the inverse projection parameters.

[0079] The encoder 30 also includes a transmitter 42 configured to transmit to a decoder an encoded bitstream including the encoded projected planes, the encoded sparse-points and the encoded projection parameters.

[0080] According to the represented embodiment, a bus 44 provides a communication path between various elements of the encoder 30. Other point-to-point interconnection options (e.g. non-bus architecture) are also feasible.

[0081] Figure 7 is a block diagram of an exemplary embodiment of a decoder 50 implementing the reconstructing method of the present disclosure.

[0082] Advantageously, the decoder 50 includes one or more processors and a memory 52.

[0083] The decoder 50 includes:

- a receiver 54 configured to receive an encoded bitstream including encoded projected planes and encoded sparse points;
- a decoding module 56 configured to decode the received encoded bitstream; and
- a reconstruction module 58 configured to reconstruct plane-like patches of the point cloud

[0084] According to the represented embodiment, a bus 60 provides a communication path between various elements of the decoder 50. Other point-to-point interconnection options (e.g. non-bus architecture) are also feasible.

[0085] While there has been illustrated and described what are presently considered to be the preferred embodiments of the present invention, it will be understood by those skilled in the art that various other modifications may be made, and equivalents may be substituted, without departing from the true scope of the present invention. Additionally, many modifications may be made to adapt a particular situation to the teachings of the present invention without departing from the central inventive concept described herein. Furthermore, an embodiment of the present invention may not include all of the features described above. Therefore, it is intended that the present invention is not limited to the particular embodiments disclosed, but that the invention includes all embodiments falling within the scope of the appended claims.

[0086] Expressions such as "comprise", "include", "incorporate", "contain", "is" and "have" are to be construed in a non-exclusive manner when interpreting the description and its associated claims, namely construed to allow for other items or components which are not explicitly defined also to be present. Reference to the singular is also to be construed to be a reference to the plural and vice versa.

[0087] A person skilled in the art will readily appreciate that various parameters disclosed in the description may be modified and that various embodiments disclosed and/or claimed may be combined without departing from the scope of the invention.

Claims

1. Method for encoding a point cloud comprising:

- segmenting (2) the point cloud into first plane-like patches;
- projecting (10) the first plane-like patches to projected planes using inverse projection parameters;
- encoding (12) the projected planes;
- determining (18) sparse points in the point cloud; and
- encoding (20) the sparse points using an octree-decomposition.

2. Method of claim 1, wherein the sparse points include points not belonging to any plane-like patch.

3. Method of claim 1 or 2, comprising:

- reconstructing (14) second plane-like patches from the encoded projected planes;
- comparing (18) the second plane-like patches and the first plane-like patches in order to obtain, for each point in the first plane-like patches, a reconstruction error; and
- comparing (18), for each point in the first plane-like patches, the reconstruction error to a threshold.

4. Method of any one of claims 1 to 3, wherein the sparse points include residual points for which the reconstruction error is larger than the threshold.

5. Method of any one of claims 1 to 4, wherein the inverse projection parameters include positions and/or orientations

of the plane-like patches.

6. Method of any one of claims 1 to 5, comprising encoding (6) the inverse projection parameters.

5 7. Method of claim 6, comprising decoding (6) the encoded inverse projection parameters.

8. Method of claim 7, wherein projecting (10) the first plane-like patches to projected planes uses the decoded inverse projection parameters.

10 9. Encoder (30) for encoding a point cloud, comprising:

- a segmentation module (34) configured to segment the point cloud into first plane-like patches;
- a projection module (36) configured to project the first plane-like patches to projected planes using inverse projection parameters;
- 15 - a first coding module (38) configured to encode the projected planes;
- a computation module (39) configured to determine sparse points in the point cloud; and
- a second coding module (40) configured to encode the sparse points using an octree-decomposition.

20 10. Encoder of claim 9, wherein the first coding module (38) is configured to encode and decode the inverse projection parameters.

11. Method for reconstructing a point cloud, comprising:

- receiving an encoded bitstream including encoded projected planes and encoded sparse points;
- 25 - decoding (22, 24, 28) the received encoded bitstream;
- reconstructing (26) plane-like patches of the point cloud.

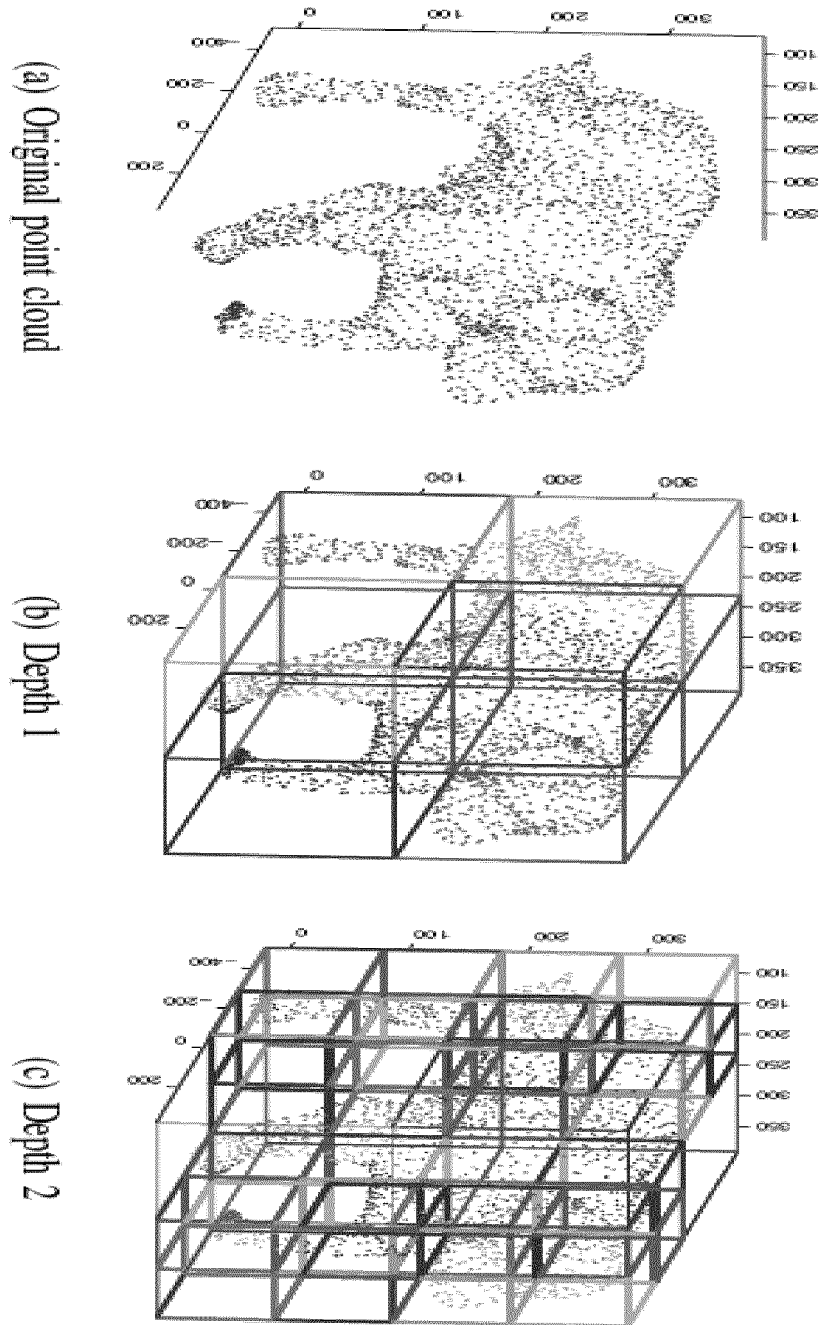
12. Method of claim 11, wherein the encoded bitstream includes encoded inverse projection parameters and the reconstructing method comprises decoding (24) the encoded inverse projection parameters.

30 13. Decoder (50) for reconstructing a point cloud, comprising:

- a receiver (54) configured to receive an encoded bitstream including encoded projected planes and encoded sparse points;
- 35 - a decoding module (56) configured to decode the received encoded bitstream; and
- a reconstruction module (58) configured to reconstruct plane-like patches of the point cloud.

40 14. A computer-readable program comprising computer-executable instructions to enable a computer to perform the encoding method of any one of claims 1 to 8.

15. A computer-readable program comprising computer-executable instructions to enable a computer to perform the reconstructing method of claim 11 or 12.



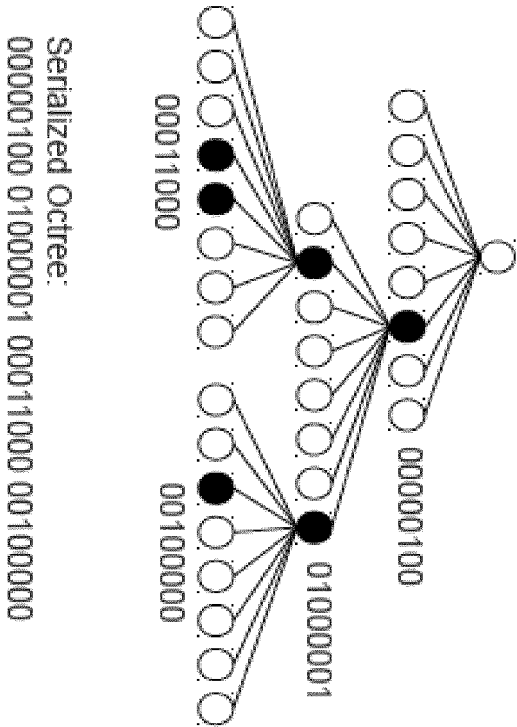
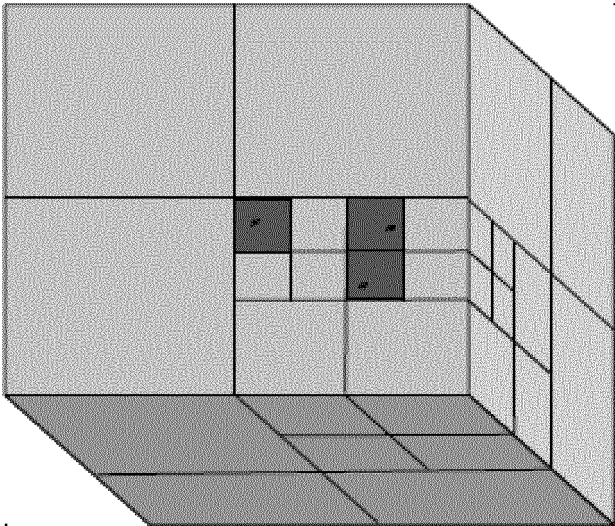


FIG.2

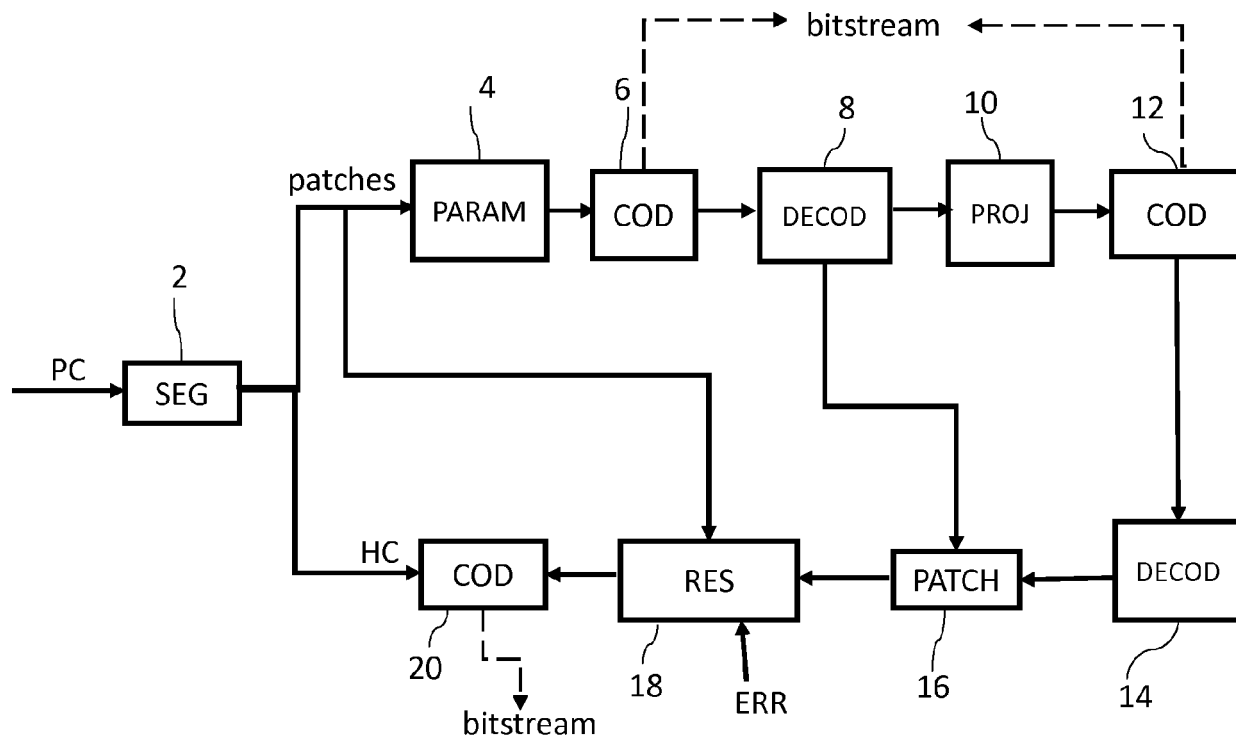


FIG.3

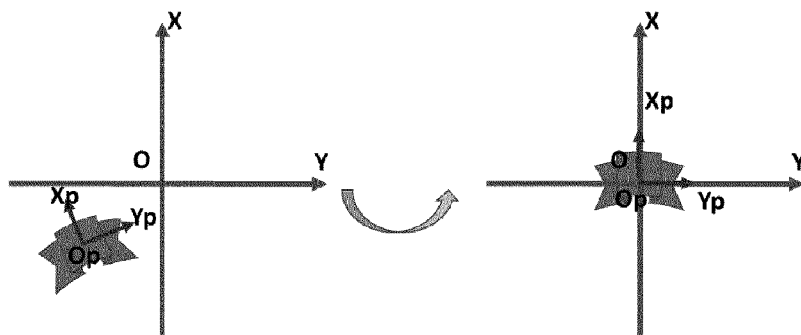
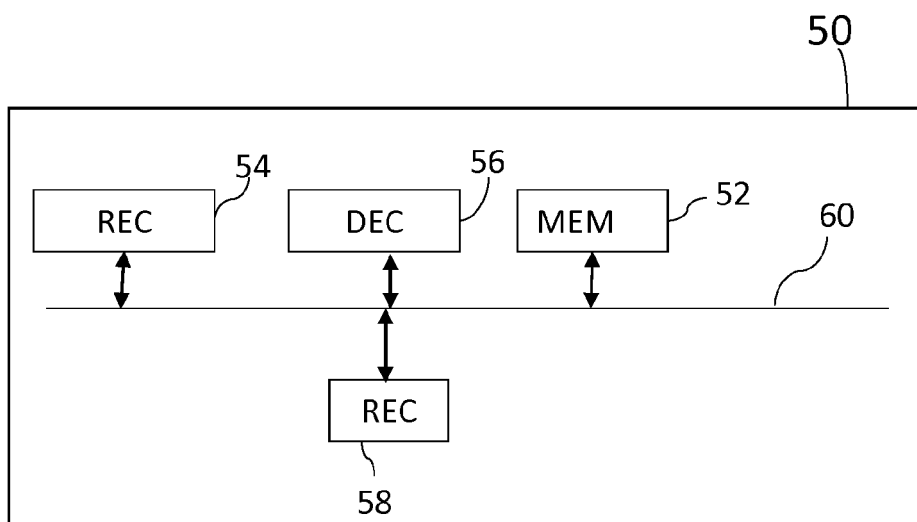
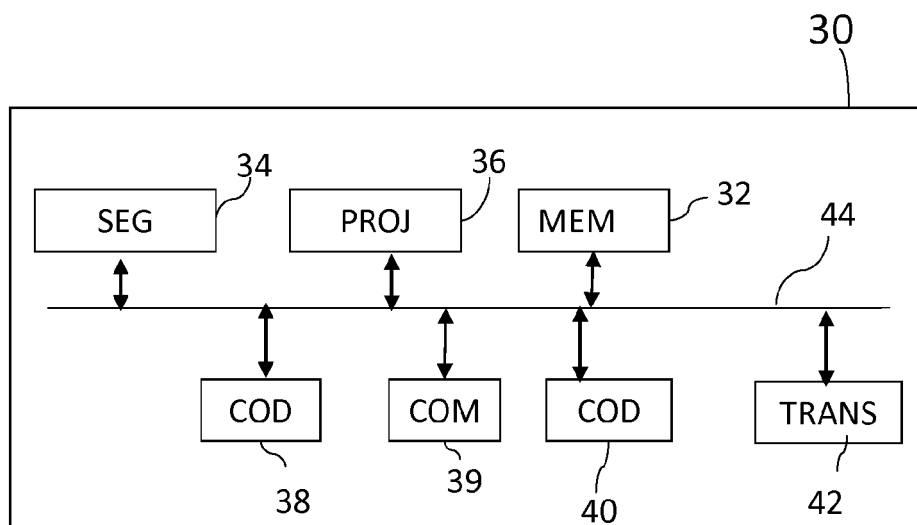
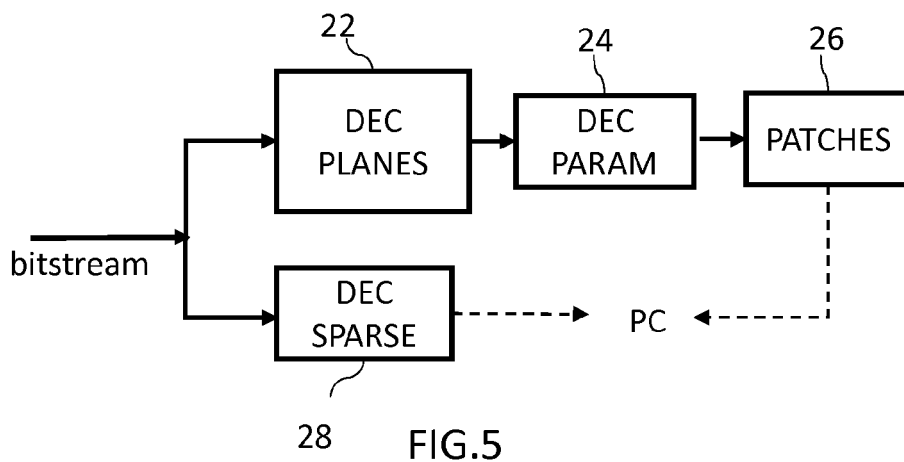


FIG.4





EUROPEAN SEARCH REPORT

Application Number
EP 17 30 5770

5

10

15

20

25

30

35

40

45

50

55

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	AINALA KHARTIK ET AL: "An improved enhancement layer for octree based point cloud compression with plane projection approximation", PROCEEDINGS OPTICAL DIAGNOSTICS OF LIVING CELLS II, SPIE, US, vol. 9971, 27 September 2016 (2016-09-27), XP060078033, ISSN: 0277-786X, DOI: 10.1117/12.2237753 ISBN: 978-0-8194-0228-8 * the whole document *	1,9,11, 13-15	INV. G06T9/40 G06T9/00 H04N19/593
X,D	T. OCHOTTA; D. SAUPE: "Compression of point-based 3d models by shape-adaptive wavelet coding of multi-heightfields", PROC. EUROGRAPHICS SYMP. ON POINT-BASED GRAPHICS, 2004, XP55397079, * the whole document *	11-13	
Y		1,5-10, 14,15	
Y,D	J. PENG; C.-C. JAY KUO: "Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition", ACM TRANS. GRAPH, vol. 21, no. 2, July 2005 (2005-07), pages 609-616, XP58335247, * abstract *	1,5-10, 14,15	TECHNICAL FIELDS SEARCHED (IPC) G06T H04N
A	TILO OCHOTTA ET AL: "Image-Based Surface Compression", COMPUTER GRAPHICS FORUM, vol. 27, no. 6, 1 September 2008 (2008-09-01), pages 1647-1663, XP55121633, ISSN: 0167-7055, DOI: 10.1111/j.1467-8659.2008.01178.x * abstract * * sections 1 and 3 *	1-15	
The present search report has been drawn up for all claims			
Place of search Berlin		Date of completion of the search 30 November 2017	Examiner Santos Luque, Rocio
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document			

EPO FORM 1503 03.82 (P04C01)

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 9214042 B [0053]

Non-patent literature cited in the description

- **J. PENG ; C.-C. JAY KUO.** Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition. *ACM Trans. Graph*, July 2005, vol. 21 (2), 609-616 [0008] [0065]
- **T. OCHOTTA ; D. SAUPE.** Compression of point-based 3d models by shape-adaptive wavelet coding of multi-heightfields. *Proc. Eurographics Symp. on Point-Based Graphics*, 2004 [0013] [0046] [0059]
- **J. DIGNE ; R. CHAINE ; S. VALETTE et al.** Self-similarity for accurate compression of point sampled surfaces. *Computer Graphics Forum*, 2014, vol. 33, 155-164 [0013]
- **J. K. N. BLODOW ; R. RUSU ; S. GEDIKLI ; E. S. M BEETZ.** Real-time compression of point cloud streams. *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 14 May 2012, 778, , 785 [0016]